

(12) UK Patent Application (19) GB (11) 2 317 723 (13) A

(43) Date of A Publication 01.04.1998

(21) Application No 9620665.1

(22) Date of Filing 30.09.1996

(71) Applicant(s)
Viewinn Plc

(Incorporated in the United Kingdom)

Sir John Lyon House, 5 High Timber Street,
LONDON, EC4V 3NN, United Kingdom

(72) Inventor(s)
Ian Roger James

(74) Agent and/or Address for Service
R G C Jenkins & Co
26 Caxton Street, LONDON, SW1H 0RJ,
United Kingdom

(51) INT CL⁶
G06F 12/08 17/30

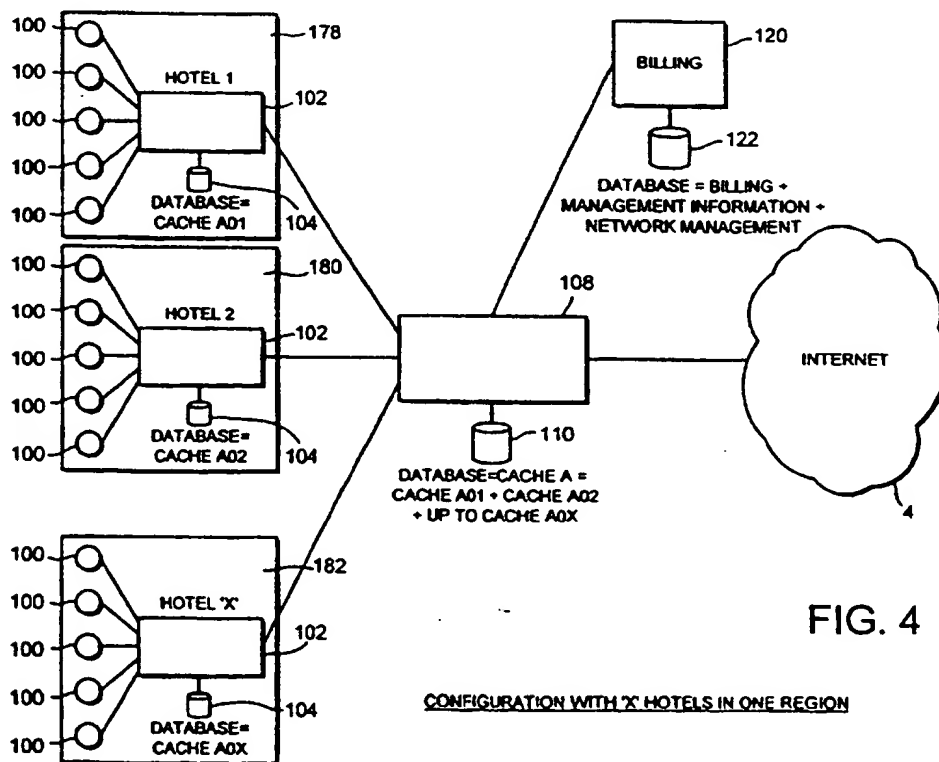
(52) UK CL (Edition P)
G4A AMC AMU

(56) Documents Cited
EP 0604013 A2 WO 95/21415 A1 US 5305389 A
US 4897781 A

(58) Field of Search
UK CL (Edition O) G4A AMC AMU
INT CL⁶ G06F 12/08 17/30
On-line databases: COMPUTER, INSPEC, WPI

(54) Caching system for information retrieval

(57) An hotel room information retrieval system, for retrieving World-Wide Web pages from the internet, includes a proxy cache server 102 in each of the hotels in the system, and one or two further levels of proxy cache servers 108 in a hierarchial structure. A client terminal 100 is located in each of the hotel rooms for access by a user. Each of the proxy cache servers stores a dynamic address list, and static address list, of Web page addresses (URLs) for which Web pages are to be pre-fetched and cached in anticipation of retrieval by a user. The dynamic address list is modified in accordance with the actual retrieval frequencies for Web pages appearing on the list, whereas the static address list is always pre-fetched.



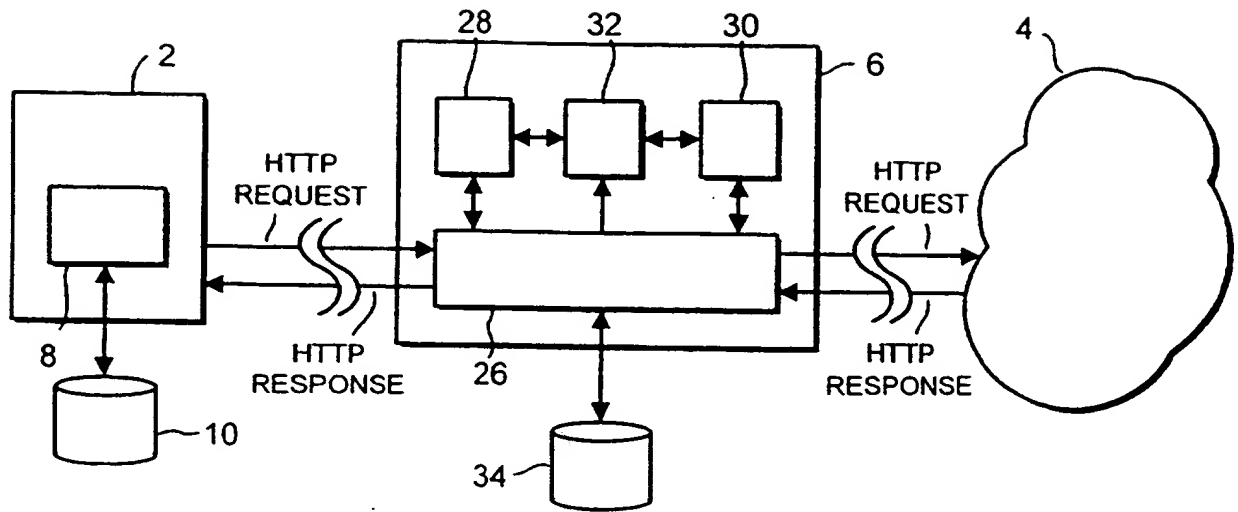


FIG. 1

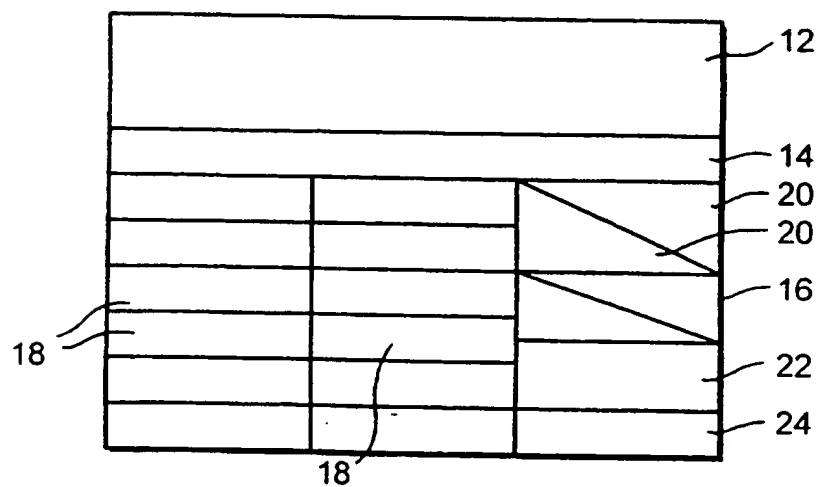


FIG. 2

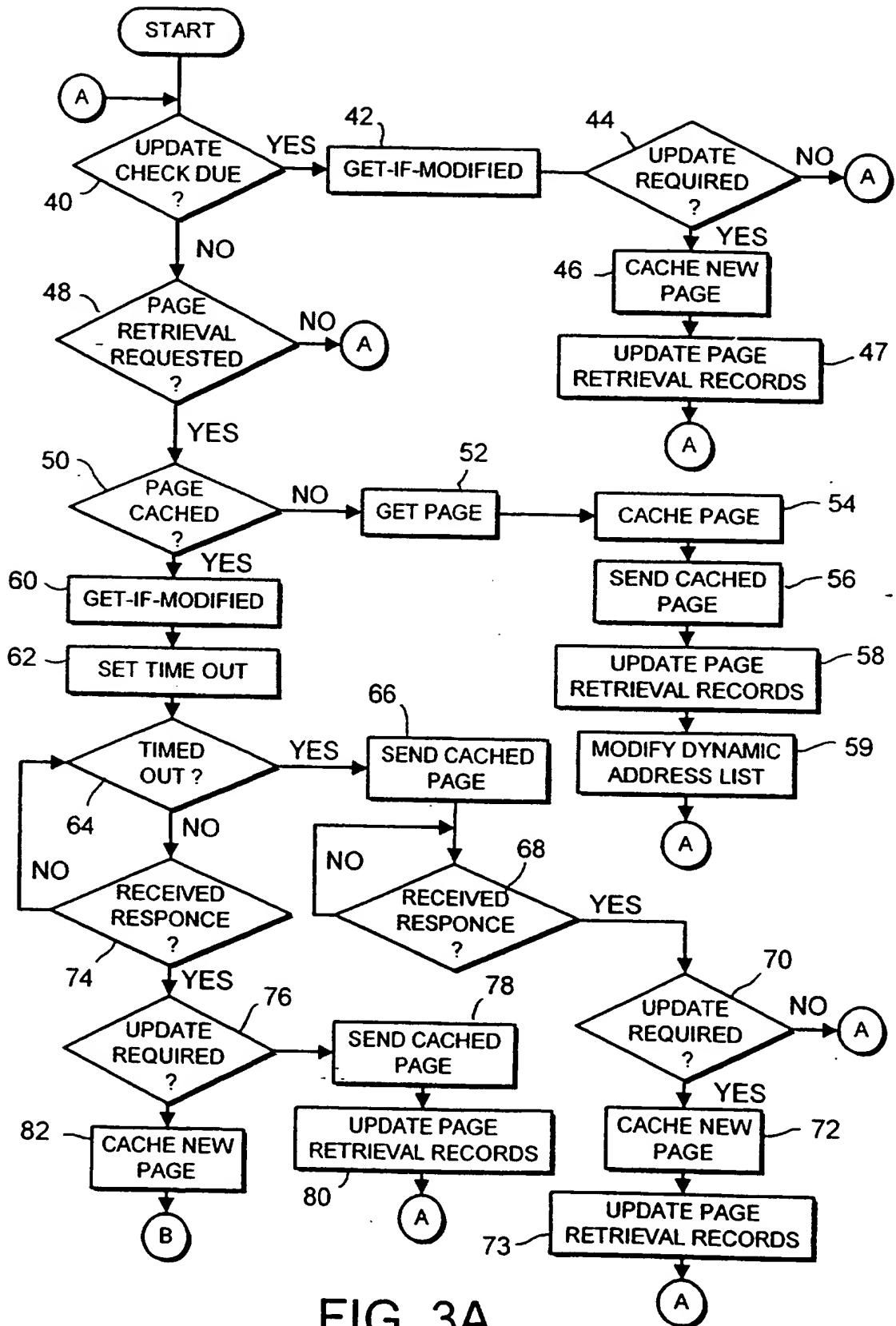


FIG. 3A

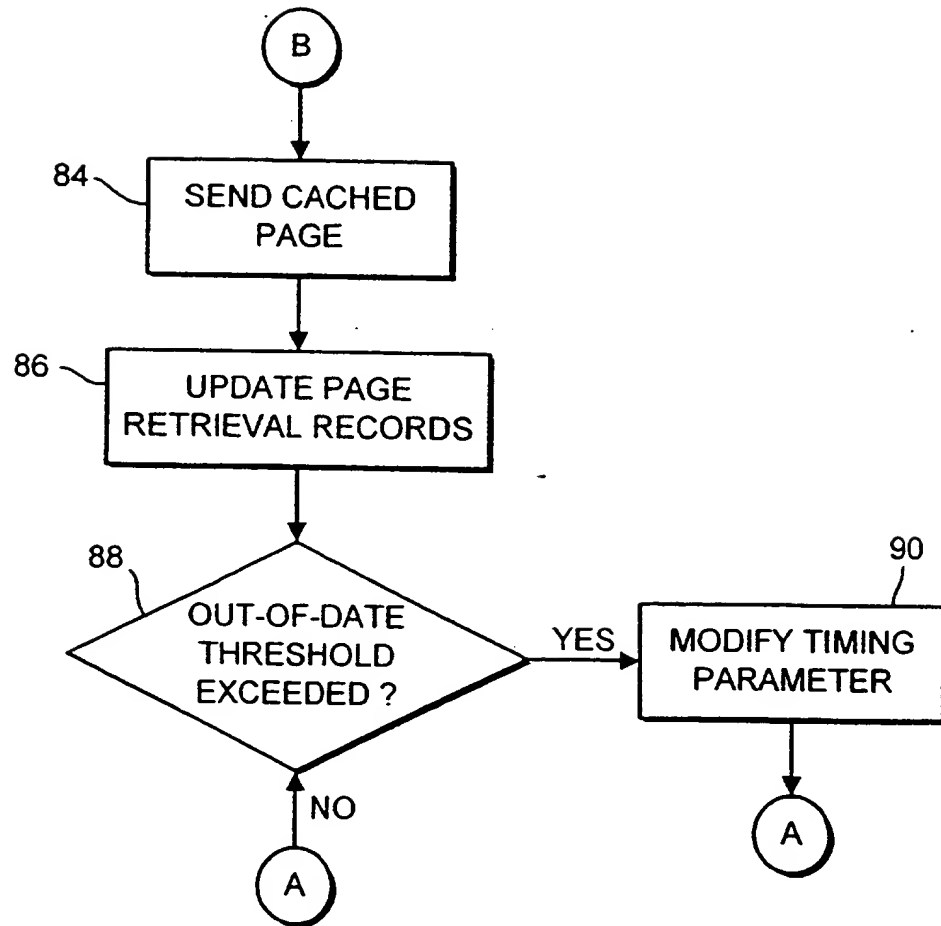


FIG. 3B

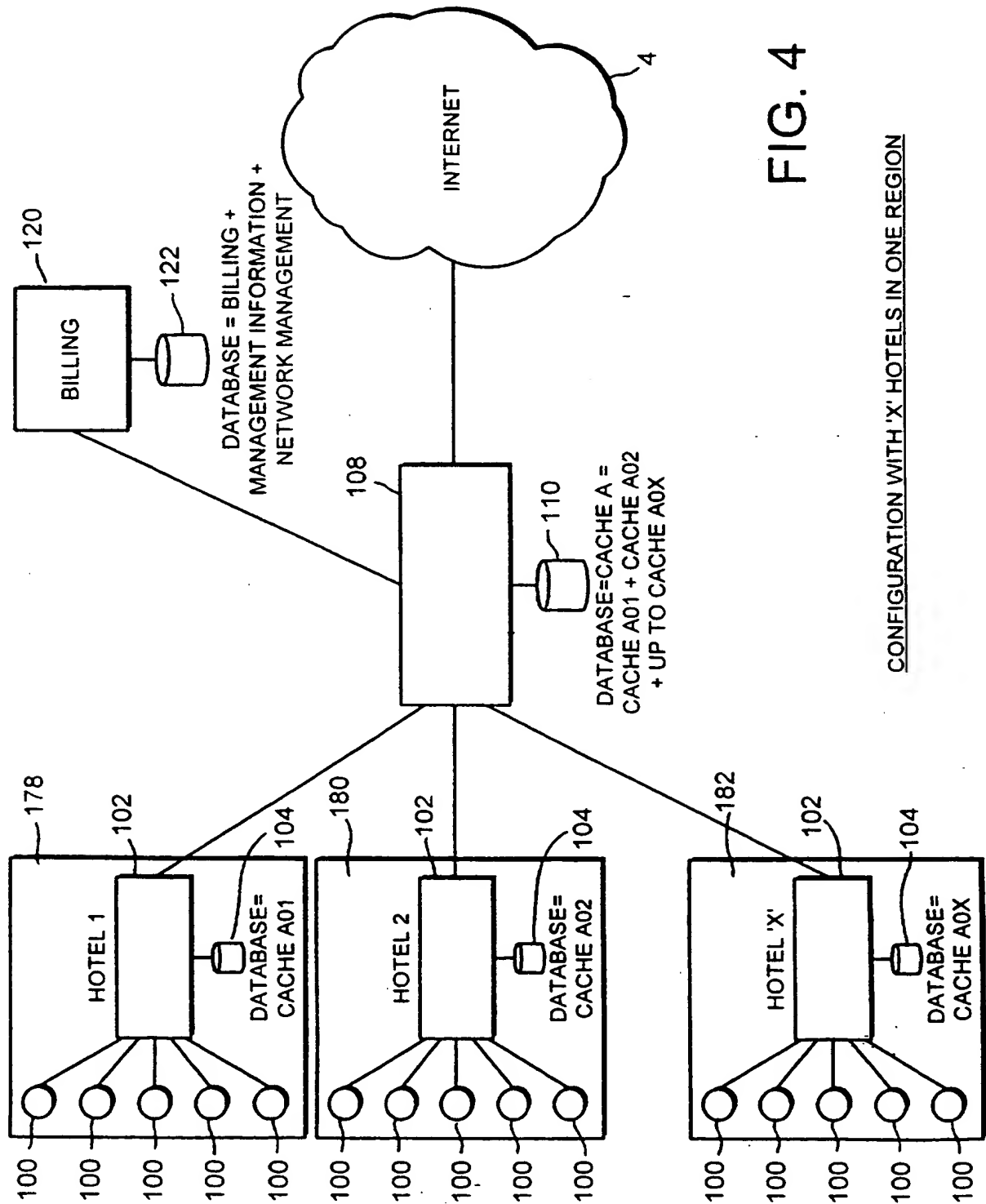


FIG. 4

CONFIGURATION WITH 'X' HOTELS IN ONE REGION

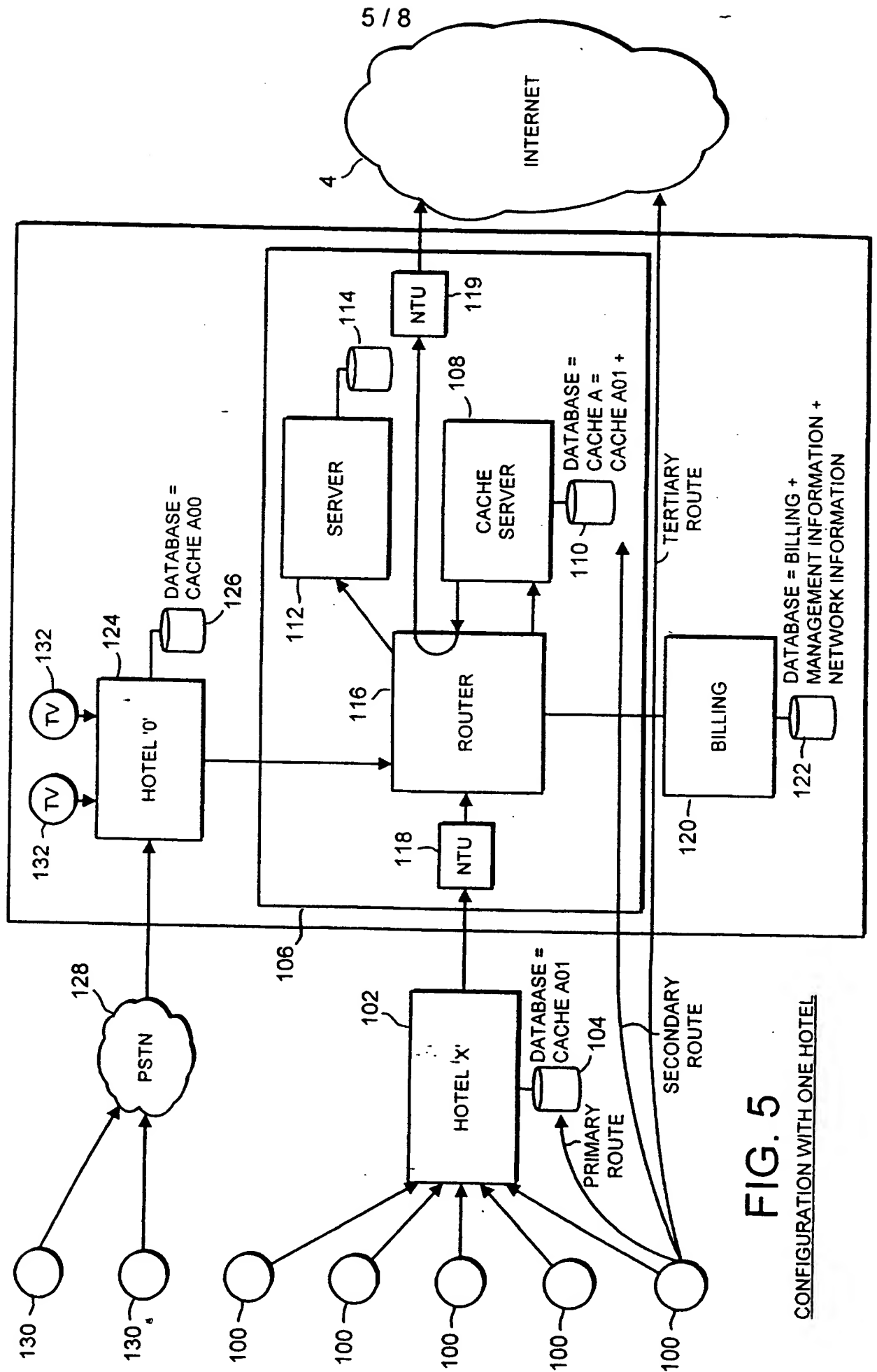


FIG. 5

CONFIGURATION WITH ONE HOTEL

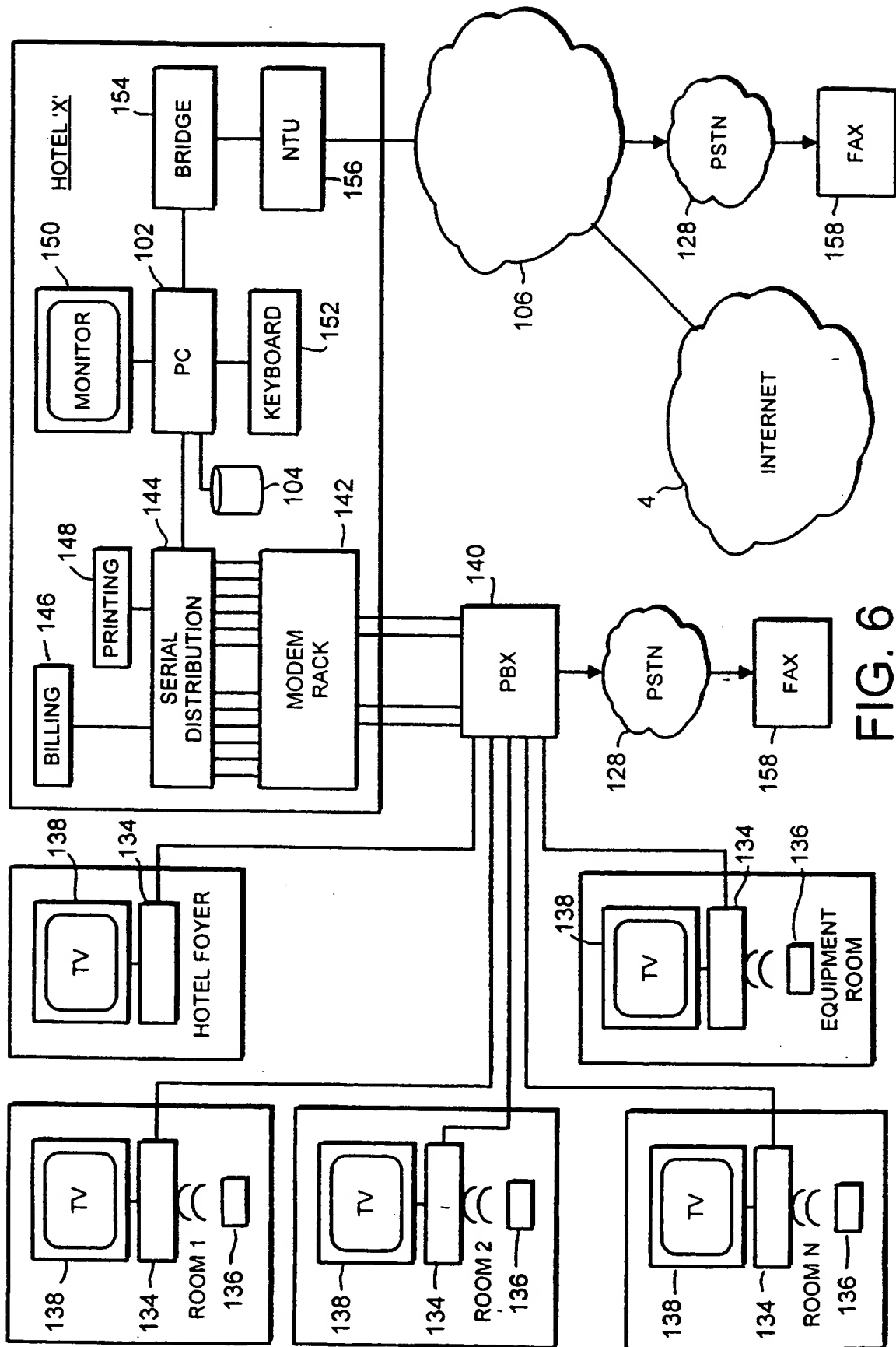


FIG. 6

HOTEL CONFIGURATION

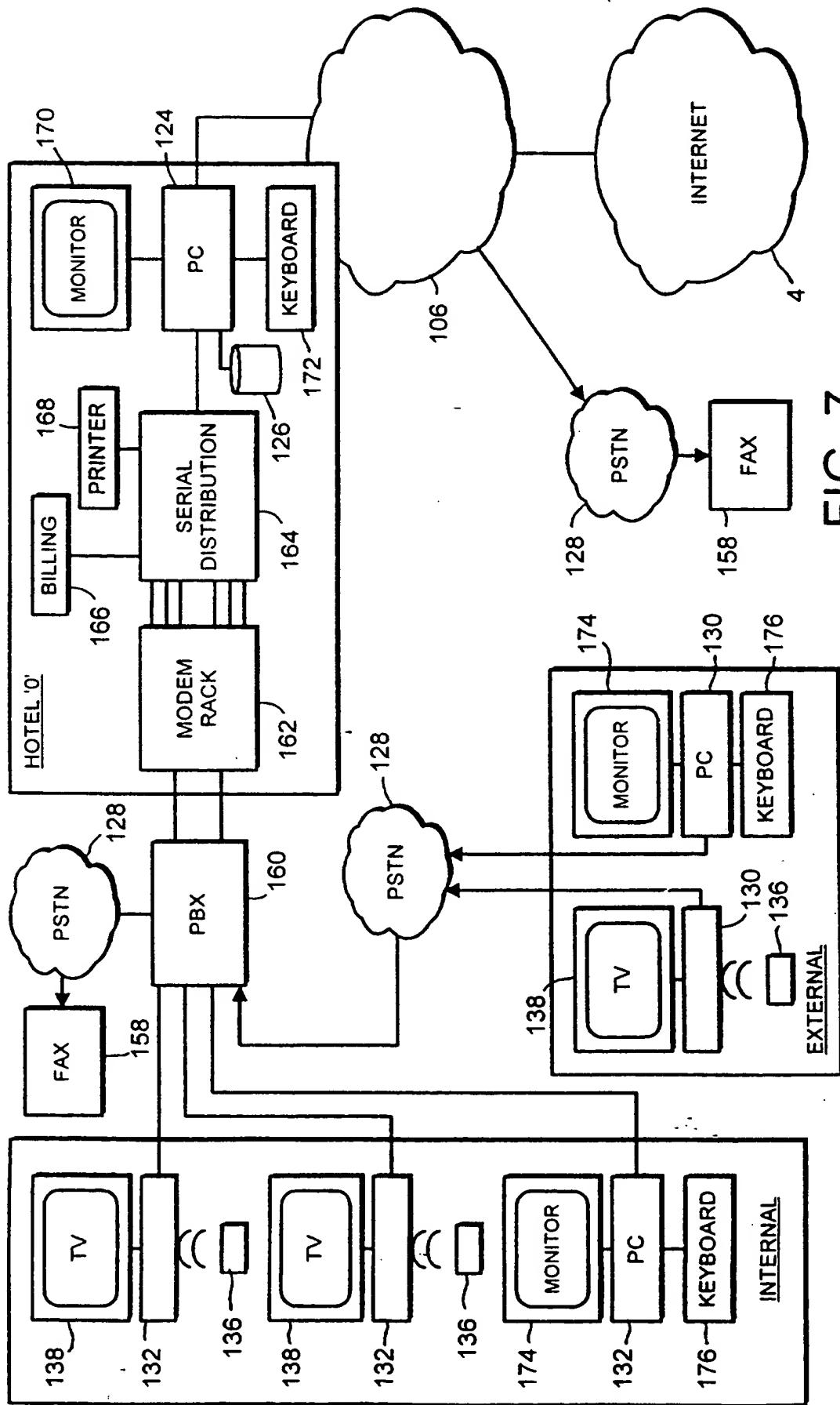


FIG. 7

HOTEL 'O' CONFIGURATION

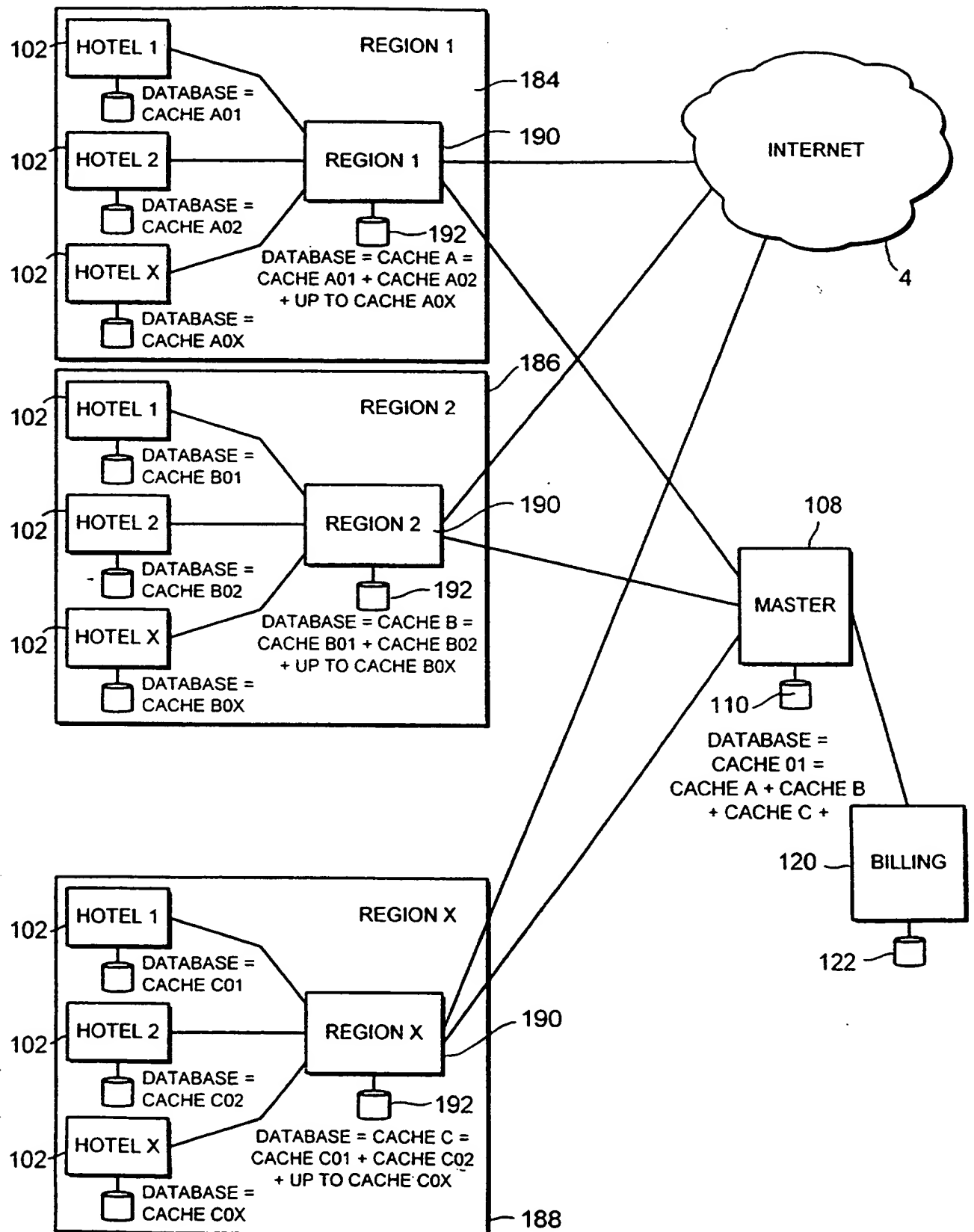


FIG. 8

CONFIGURATION WITH 'X' HOTELS IN 'X' REGIONS

CACHING SYSTEMS

This invention relates to caching systems for requesting data files from a server network, in which said data files are identified by file addresses, and
5 for storing same prior to retrieval via said system by a client.

The invention relates in particular, but not exclusively, to caching systems for pre-fetching and caching World-Wide Web pages from the internet.

10 Caching systems for caching Web pages are in themselves known. In one known system, when a Web page is retrieved from a Web site, a proxy cache server caches the retrieved Web page during transmission to the requesting party. This allows the
15 proxy cache server to provide the cached Web page if subsequently requested again, without the need to return to the original Web site. A drawback however is that the initial retrieval of a Web page must still be made in real-time from the internet, and this can
20 result in variable and sometimes protracted delays depending on the current load of the internet links used.

A further known caching system is that which uses the functionality provided by the HTTP protocol, in
25 which an "expires: date" header field may be sent with a Web page when retrieved from the Web site. This

command indicates the day after which the Web page ceases to be valid and should be retrieved again. Thus, once a Web page has been cached, a known caching system re-fetches the contents of the page at the same address, after the expiry date of the previous page. However, many documents do not have an expiry date specified when retrieved and therefore this HTTP functionality is limited in utility.

In accordance with one aspect of the present invention there is provided a caching system for requesting data files from a server network, in which said data files are identified by file addresses, and for storing same prior to retrieval via said system by a client, said system being configured to:

- (i) access a dynamic set of file addresses;
- (ii) request and store data files having file addresses appearing in said set;
- (iii) assess the likelihood of retrieval of a data file via said system in accordance with predefined criteria; and
- (iv) modify said set in accordance with said assessed likelihood.

An advantage of this arrangement is that a dynamic set of file addresses, for which pre-fetching is to be performed, can be defined, and the set of file addresses can be dynamically modified in

accordance with the likelihood of retrieval of a data file, assessed in accordance with predefined criteria. This allows file addresses to be added and removed from the set of pre-fetched files, to adapt the sets of pre-fetched files to estimated future requirements.

In accordance with a further aspect of the invention there is provided a caching system for requesting data files from a server network, in which said files are identified by file addresses, and for storing same prior to retrieval via said system by a client, said system being configured to:

- (i) automatically request a current version of a data file for a file address appearing in a predefined set of file addresses, for which address a data file was already stored, without reference to said stored data file;
- (ii) request a current version of a data file for said file address when being retrieved by a client; and
- (iii) store said current version when received from said server network, for subsequent retrieval by a client.

Thus, a version of a data file can be pre-fetched automatically, without reference to an "expires: date" HTTP header field. Even when pre-fetched, the version

may not be current when subsequently being retrieved by a client, in which case the current version may be requested. Thus, in all cases, it can be ensured that the current version of the data file is retrieved by the client, whilst in general the retrieval access time can be minimised by virtue of the pre-fetching characteristic of the caching system.

In accordance with a yet further aspect of the present invention, there is provided a caching system for requesting data files from a server network, in which said files are identified by file addresses, and for storing same prior to retrieval via said system by one or more clients, said system being configured to automatically request a current version of a data file for a file address appearing in a predefined set of file addresses, said set comprising members selected in response to the retrieval of corresponding data files via said system, and members selected independent of retrieval of a corresponding data file via said interface.

This allows a set of file addresses for which a file is always to be pre-fetched to be defined, along with a set of file addresses which are to be pre-fetched as a result of the file addresses having recently been retrieved.

In a still further aspect of the present

invention there is provided a caching system for requesting data files from an external server network, in which said data files are identified by file addresses, and for storing same prior to retrieval via said system by a client, said system comprising a plurality of low-level cache servers, each said low-level cache server respectively having an associated plurality of clients, and a higher-level cache server serving said plurality of low-level cache servers, said system being configured such that each of said low-level cache servers stores an individual set of data files, and said higher-level cache server stores data files appearing in each of said individual sets.

This provides an improved functionality in cache server systems. By storing data files appearing in each of the individual sets of the low-level cache servers, the higher-level cache server is able to provide a super set of data files for retrieval by a client in any of the low-level cache server networks served by the higher-level cache server.

Another aspect of the present invention provides a caching system for requesting data files from a server network, in which said data files are identified by file addresses, and for fetching and storing same prior to retrieval by a user, said system comprising user profile storage means, accessible by

the system, said user profile comprising a set of data file addresses to be fetched and stored by said system for subsequent retrieval.

Thus, the usual access that a user requests when
5 accessing the caching system can be pre-fetched automatically into a cache when the profile is accessed by the system. This can be used to improve the response time when the user wishes to receive any of the data files appearing on the user profile.

10 The present invention provides for the pre-caching of a defined set of Web pages in anticipation of subsequent retrieval of those Web pages by a user, and for the modification of the set of Web pages to be pre-fetched in accordance with an assessed likelihood
15 of such retrieval.

Embodiments of the present invention will now be described, by way of example only, with reference to the accompanying drawings in which:

Figure 1 shows a block diagram illustrating
20 principles of embodiments of the invention;

Figure 2 shows a menu page displayed by the client of a data retrieval system in accordance with the present invention;

Figures 3A and 3B show a flow diagram associated
25 with the block diagram of Figure 1;

Figure 4 shows a block diagram of an information

retrieval system in accordance with one embodiment of the present invention;

Figure 5 shows a block diagram of a portion of the system illustrated in Figure 4, in greater detail;

5 Figure 6 shows a block diagram of a portion of the system of Figure 5, in greater detail;

Figure 7 shows a block diagram of a different portion of the system in Figure 5, in greater detail; and

10 Figure 8 shows a further embodiment of the present invention.

Figure 1 is a block diagram illustrating client/server functionality of embodiments of the present invention. In these embodiments, the client
15 2 is configured to retrieve data files, originating within the internet 4, via a proxy cache server 6. In particular, the data files are World-Wide Web, (referred to herein as "Web") "pages". The internet itself includes a large number of Web servers
20 interconnected by data communication links, communicating using, for example, the TCP/IP communications protocol. In the case of Web pages, requests and responses are made in accordance with the Hypertext Transfer Protocol (HTTP).

25 The client 2, which may be in the form of a conventional work station such as a PC, includes a

browser application 8 for retrieving Web pages by means of transmitting HTTP requests and receiving HTTP responses. The browser also converts the Web pages when received in the Hypertext Mark-up Language (HTML) into a form suitable for display to the user. There are a number of publicly-available browsers. In this instance a Netscape browser, or any other suitable browser, may be used. The client 2 is also provided with a non-volatile data store 10, such as a hard drive, in which a number of menu pages, in HTML format, and a corresponding index of Web page addresses (known as Universal Resource Locators, or URLs) are pre-stored.

Referring to Figure 2, one of the menu pages stored in the client store 10 is in this embodiment configured as illustrated. This is the main menu page for access to the information retrieval system, which is displayed to the user when first logging on. The menu page includes a top portion 12 showing a static image, and a middle portion 14 showing or static or scrolling image (which may be implemented using the Jarva programming language), and a lower portion divided into selectable icons displayed in the form of image map.

A number of the icons 18 identify newspaper Web pages, and these icons have associated URLs stored in

the index mentioned above. Other icons 20 indicate other menu pages, also stored in data store 10, which are selectable from the main menu page. These other menu pages may be configured in a similar fashion, with icons displayed as an image map and having associated URLs referenced in the index stored in the client store 10. The other menu pages may include a "business" menu page, a "sport" menu page, a "what's on ..." menu page, a "weather" menu page, a "travel" menu page and a "magazines" menu page. Also included is a "World-wide Web" selectable icon 22, which when selected allows the user to enter a conventional browsing mode whereby the entire available information on the World-wide Web may be accessed. Also included is a "ring-off" icon 24, to allow the user to log off the information retrieval system.

In each case, the icons are selectable by means of a user input device, such as a trackball, mouse or keyboard, connected to the client 2.

The proxy cache server 6 may be implemented by use of any suitable known server, such as a Bull Power PC 604 Estrella. It includes caching application software 26, a stored dynamic address list 28, a stored static address list 30 and stored page retrieval records 32. Also provided is a cache store 34 which provides for the mass storage of Web pages.

The dynamic address list 28, the static address list 30 and the page retrieval records 32 may be stored in a separate storage device, or may be stored in common with the cache store 34. The caching application 26
5 may be any which adapts the proxy server 6 to process HTTP responses and requests, and to cache Web pages, such as the "Squid" or "Spinner" software which is available without charge on the Web.

The static address list 30 includes all, or at
10 least some of, the Web page addresses indexed in the client store 10, corresponding to the page icons displayed in each of the menu pages stored in the client store. This static address list is a set of page addresses (URLs) from which pages are always to
15 be pre-fetched into the cache server 6. The static address list 30 also stores timing parameters for each address, defining the time and/or frequency with which the associated Web page is to be pre-fetched.

The dynamic address list 28 has a form similar to
20 the static address list 30, and includes Web page addresses which are not indexed in the client store 10. When a user retrieves a Web page using the Web browsing function provided by the browser 8, the corresponding page address is added to the dynamic
25 address list 28. This dynamic address list 28 is a set of page addresses (URLs) from which Web pages are

to be pre-fetched into the cache server, as long as the Web pages are retrieved relatively frequently by a user of the system. The dynamic address list 28 also includes timing parameters associated with each address, to define the time and/or frequency of pre-
5 fetching of the corresponding Web page. When a Web page address is first added to the dynamic address list 28, a default timing parameter is associated with the new address. However, the timing parameters held
10 in address lists 28 and 30 are modifiable, as will be discussed below.

The page retrieval records include the time of receipt of the page copy, the "Web site" from which the page copy has been retrieved, the address (URL) of
15 the page, the size of the page, and an indication of whether the page, when retrieved by a user, was in up-to-date form, or whether a new page needed to be cached when retrieved.

Reference is now made to Figure 3, which is a flow diagram illustrating the procedures followed by
20 cache server 6 under the control of caching application software 26. It is to be understood that cache server 6 is in a constantly active state, to provide a 24 hour daily service, and that the start of
25 the procedure illustrated in Figure 3 may be at any randomly chosen time.

In a first step 40, the cache server 6 checks the timing parameters stored in the address lists 28 and 30 to determine whether an update check on any of the corresponding page addresses is due. If so, a pre-
5 fetch command (a "get-if-modified" HTTP demand) is sent to the internet 4. The "get-if-modified" command includes the time stamp of the presently cached page at that address. This time stamp is checked by the Web file server holding the page (the "Web site") in
10 the internet 4, to determine its response. If the page has not been modified since the Web page was originally cached from that address, the Web file server responds with a "not modified" response, in which case the cache server 6 determines in step 44
15 that no update is required, and returns to step 40. Otherwise, the Web file server transmits the new Web page data to the cache server 6, which caches the new Web page in cache store 34, step 46, updates the page retrieval records, step 47, and returns to step 40.

20 When not performing an update check, the cache server 6 checks whether a page retrieval request has been received from client 2, step 48. If not, the procedure loops back to step 40. If a page retrieval request has been received, the cache server 6 checks
25 whether it holds a copy of a Web page having the specified address in the cache store 34, step 50. If

no copy is held, the cache server 6 proceeds to issue an HTTP "get" command, specifying the page address, step 52. This results in the corresponding Web file server in the internet 4 transmitting the Web page data to the cache server, which proceeds to cache the data in cache store 34, step 54. Also stored with the cached page copy is the time stamp of the newly pre-fetched copy, which specifies the time of receipt of the copy.

10 The cached file is then transmitted, in HTTP format, to the client, step 56, and the file retrieval data 32 is updated, step 58.

15 The file retrieval data includes the page address retrieved, and the time stamp of the page copy. Since the Web page is a newly-cached Web page, the Web page address (URL) is added to the dynamic address list 28, step 59. A default timing parameter is associated with the new address list. This timing parameter may indicate, for example, that a page is to be pre-fetched (step 42) from the internet 4 once daily at an appropriate time. This is particularly true for newspaper Web pages, which are generally updated once daily. In particular, the default pre-fetching time may be set at a time before, say, 6.00 a.m., such that an updated page is available to a user in the morning.

25

 If a page at the specified page address has

previously been cached, the cache server 6 checks whether the cached page is a current version, by issuing a "get-if-modified" command, step 60. At the same time, the cache server 6 sets a time out, for example 9 seconds, in case a response to the "get-if-modified" command is not received within an acceptable time period, step 62.

If the time out period elapses before a response is received, step 64, the cache server transmits to the client 2 the presently cached page, step 66. The transmitted page is transmitted with an associated "expires: date" HTTP command, specifying a time shortly after the Web page is sent, to ensure that the client 2 will re-request a Web page at the same address within a short period, in case an updated Web page is received within that period. The cache server then proceeds to await a response, step 68. If a response is received, it may be a "not modified" response, in which case no further action is taken, step 70. Otherwise, the new page is cached along with a time stamp indicating a time of receipt, step 72, and the page retrieval records 32 are updated, step 73. The cache server then returns to step 40.

If within the time out period a responses is received, step 74, the cache server 6 determines whether the response is a "not modified" response,

step 76. If so, the cache server 6 sends the previously cached page to the client 2, step 78. The cache server 6 then updates the page retrieval data, step 80, including in this case an indication that the page, when requested by the user, was cached in an up-to-date form.

If instead the response received from the Web site is new Web page data, the cache server 6 caches the new Web page data along with a current time stamp, step 82. The newly cached page is then transmitted to the client 2, step 84. Next, the page retrieval records 32 are updated with, amongst other data, an indication that, in this case, the previously cached page was out-of-date.

The up-to-date and out-of-date data in the page retrieval records 32 is used to modify the pre-fetching timing parameters associated with the corresponding Web page. If the number of out-of-date retrieval records exceed a predetermined threshold for a particular page address, which threshold may be as low as one, the timing parameter associated with the page address is modified to increase the pre-fetching frequency, thereby to reduce a likelihood of the cached page being out-of-date, steps 88 and 90. The cache server 6 then returns to step 40.

The page retrieval records 32 are also used in a

procedure for deleting infrequently accessed Web page addresses from the dynamic address list 28. Thus, the caching application 26 determines, by means of the page retrieval records 32, whether each of the Web page addresses held in the dynamic address list 28 has been recently accessed, within a predetermined period, example the last 72 hours, i.e. three days. If not, the address is removed from the dynamic address list, thereby to reduce the caching load for the information retrieval system. It is assumed that, since the Web page address in question has not recently been accessed within the time period specified, the user may well have left the hotel and the likelihood of further access being required to that address in the near future is much reduced.

Other embodiments of the present invention will now be described with reference to Figures 4 to 9. Each of these embodiments relates to an hotel room internet service.

Figure 4 shows a basic schematic illustration of an information retrieval system in accordance with an embodiment of the invention, in which internet access is provided in the rooms of a plurality of hotels including hotel 1, 178, hotel 2, 180, up to and including hotel 'X', 182. Each of the hotels is provided with an hotel cache server 102 and associated

client terminals 100 located in the hotel rooms. Each of the hotel cache servers 102 is connected to a central cache server 108 via ISDN lines. Both the hotel cache servers 102 and the central cache server 108 are configured in accordance with the arrangement described in relation to Figures 1 and 3A, 3B, such that each has the pre-fetching and caching functionality provided. Each of the hotel cache servers 102 has a cache data store 104, whereas the central cache server 108 has a larger capacity data store 110. A billing server 120, monitoring usage of the system, and provided data store 122, is connected to central cache server 108.

Figure 5 shows further details of the system illustrated in Figure 4. Only one hotel is illustrated for simplicity, however the arrangement in each hotel is essentially similar. A plurality of client terminals 100, each located in a separate hotel room, are connected, via internal telephone line connections to a cache server 102, with an associated cache store 104, located in the hotel basement. The hotel cache server 102 is in turn connected, via an ISDN link, to a master system 106, which is remotely located. The master system 106 includes a cache server 108, an associated cache store 110, a Web server 112 and its associated data store 114, a router

116 and network terminal units, comprising high-speed modems 118 and 119 for connecting the master system to the hotel server 102 and the internet 4.

5 Both the hotel cache server 102 and the master cache server 108 are configured in accordance with the proxy cache server format described above in relation to Figures 1 and 3, and the advantages of a hierarchial cache server structure will become clear when the provision of a plurality of hotel cache
10 servers is explained further below.

Also provided is a billing file server 120, and its associated data store 122, connected to the master system router 116. Co-located with the master system is a further cache server 124 (the "hotel '0' cache
15 server") also configured in the fashion of the cache server 6 described in relation to Figures 1 and 3 along with its associated cache store 126. The further cache server 124 is provided primarily for remote access via a public services telephone network
20 (PSTN) 128, by remote client terminals 130. Local client terminals 132 also have access to the information retrieval system via the remote access cache server 124.

Figure 6 illustrates the portion of the
25 information retrieval system located at the hotel. A number of the hotel rooms (rooms 1 to n) are each

provided with a client terminal 134 with a user input device 136. The user terminal 134 is configured in the format of the client 2 illustrated in Figure 1. Each user terminal is connected to the hotel room TV 138 already present in the hotel room.

Further user terminals 134 may be provided in the hotel foyer and in the hotel equipment room, for use by visitors and staff.

The user input devices 136 may communicate via infra-red communications, to increase the convenience to the user. In particular, the user terminals 134 may be in the form of an Envision PC, manufactured by Olivetti, which are provided with a keyboard with an infra-red connection with the PC.

Each of the user terminals 134 is connected via telephone lines and the hotel PBX 140, already provided in hotels, to a modem rack 142. The modem rack 142 is connected to a serial distribution link 144, which provides signals to the hotel cache server 102 in serial form. A billing terminal 146 is connected to the serial distribution link 144 to log connection times for billing purposes. A printer 148 is connected to the serial distribution link 144 to allow the printing of Web pages in response to a command sent by a terminal unit 134.

The cache server 102 is provided with a monitor

150, a keyboard 152 and its cache store 104. The
cache server 102 is linked in turn, via a network
bridge 154 and a network terminal unit 156, to the
master system 106, which as previously indicated is in
5 turn connected to the internet 4.

The network terminals 134 are configured such
that a fax may be generated by a user and sent, either
from the hotel PBX 140, or from the master system 106,
via the PSTN 128 to a remote fax station 158.

10 Figure 7 shows the portion of the information
retrieval system shown in Figure 5 relating to the
external access cache server 124, in greater detail.
User terminals 130 not resident in an hotel are
connected via the PSTN 128 to a PBX co-located with
15 the external cache server 124. The PBX 160 is
connected to a modem rack 162, which in turn connects
via a serial distribution link 164 to the cache server
124. A billing terminal 166 and a printer 168 are
connected to the serial distribution link 164. A
20 monitor 170 and a keyboard, along with the cache store
126, are connected to the cache server 124. The cache
server 124 is in turn connected to the master system
106, as indicated in Figure 4. Also connected to the
PBX 160 via internal telephone lines are client
25 terminals 132, which may be in the form of a
conventional PC and/or Envision PCs. Thus, the

display unit may be in the form of a TV 138 or a monitor 174, and the user input terminal may be in the form of an infra-red input device 136 or a conventional computer keyboard 176. The same applies to the client terminals 130 which are externally located.

The external access system is configured to allow fax messages generated in client terminals 130 or 132 to be sent to remotely located faxes 158, via the PSTN 128, from either PBX 160 or from the master system 106.

Figure 8 shows a further embodiment of the present invention, in which a number of hotel cache servers 102, each located at a different hotel, in different regions 184, 186 and 188 are connected to regional cache servers 190, provided with regional cache stores 192. Each of these regional cache servers is also configured in accordance with the cache server described in relation to Figures 1 and 3, such that each stores a dynamic address list, a static address list and page retrieval records. In addition to each regional cache server 190 being connected to the master server 108 via ISDN links, each regional server has a direct ISDN connection with the internet 4, as illustrated in Figure 8.

The hierarchical cache server networks illustrated

in Figures 4 to 8 have the advantage of reducing the need for direct and real-time access to the internet 4 when a user retrieves a page, whilst ensuring an efficient caching mechanism. A static address list 30
5 appropriate to the status of each cache server can be prestored in each cache server.

The static address list for each of the hotel cache servers 102 includes Web page addresses considered to be likely to be accessed according to an
10 individual customer profile of each hotel. For example, if an hotel has a predominant Japanese clientele, the static address list will include a larger proportion of Japanese newspaper Web page addresses. The client store 10 in each client
15 terminal 100 stores a corresponding menu containing corresponding Japanese newspaper titles, and an associated index of the Web page addresses stored in static address list 30.

The regional cache servers 190 on the other hand
20 are provided with a static address list 30 of national and regional news and sport Web page addresses, which are to be pre-cached in the regional cache server 190. These are Web pages which are considered to be likely
to be accessed by users within that region, and the
25 pre-fetching of the Web pages into the regional cache store 192, which can be performed during low usage

periods of the system, reduces the data communications load between the regional cache server 190, the internet 4 and/or the master cache server 108, during relatively high usage periods.

5 The static address list 30 and the master cache server 108 can meanwhile contain addresses for international news and sport pages, likely to be accessed by users across the information retrieval system.

10 As will be appreciated when considering the functionality of the proxy cache server 6, which each of the cache servers 102, 190 and 108 adhere to, when a regional cache server 190 requests and receives Web page data via the master cache server 108, the
15 received data is cached in both the regional cache store 192 and the master cache store 110. Thus, a Web page pre-fetched or otherwise retrieved in one region is then readily available for retrieval or pre-
20 fetching by a different region, in master data store 110. Similarly, a Web page cached in an hotel cache store 104 will also be cached in the corresponding regional cache store 192 and the master cache store 110. The cached Web page is therefore available from
25 the regional cache store to the remaining hotel cache servers 102 in the same region, and from the master cache store 110 to the remaining hotel cache servers

102.

In each case when an hotel cache server 102, a regional cache server 190 or the master cache server 108 receives a "get-if-modified" command from a lower-level cache server, it re-transmits a "get-if-modified" command using the time stamp of the page copy cached in its own data store. Thus, if an up-to-date copy is stored in the regional cache store 192 or the master cache store 110, whilst an out-of-date copy is stored in the hotel cache store 104 or the regional cache store 192, a new copy need not be retrieved from the internet since the copy will be provided from the regional cache store 192 or the master cache store 110 as appropriate.

Each of the hotel cache stores 104 also stores self-authored Web pages, accessible via the menus and associated indices stores in the client store 10, which provide local information. These self-authored pages include pages detailing the hotel room facilities, other facilities of the hotel, functions and conferences, details of the immediate surroundings, for example restaurants, parks, etc., and travel connections from the hotel environs.

The master cache server 108 may also hold within its static address list Web page addresses specified as preferred by frequent users of the information

retrieval systems. Therefore, whichever hotel or region a frequent user accesses the system from, their preferred Web pages are pre-fetched and stored at least in the master cache store 110.

5 The master cache server may also be configured to function as a mail server in addition to as a cache server, to allow users of the information retrieval system to retrieve E-mail messages at the client terminals 102 located within the hotels, or at the
10 external client terminals 130 connected to the system via PSTN lines.

 In a further addition to the information retrieval system of the present invention, the client terminals 102, 130 and 132 may be configured to allow
15 an individual user to define a preferred user access profile when logged on. In accordance with this additional feature, the user is provided with a smart card (containing a microprocessor and a non-volatile memory), which is insertable within a smart-card
20 reader on the client terminal.

 The user identity is first verified by means of a personal identification number (PIN) input via the user input device, the PIN being stored in encrypted form on the smart card. Once the PIN is verified, the
25 user may specify preferred Web page addresses to be accessed when visiting an hotel.

When the user subsequently logs on at a hotel client terminal 102, or any other user terminal, the user terminal initiates an automatic pre-fetching procedure for each of the specified Web page addresses in the user access profile held on the smart card. The user is presented with a display screen showing each of the Web page addresses in the user's access profile. As each of the corresponding Web pages is pre-fetched, an icon adjacent the corresponding Web page address is altered to indicate that pre-fetching for that Web page address is complete (the page being cached at each level of the system). An additional icon is also provided on this personal menu page to indicate whether the user has yet accessed the contents of the Web page in the same user session.

In addition, the client terminal 102, 130 or 132 may be configured to allow the user to define a preferred screen presentation format when accessing the system in future. Thus, the user may customise the presentation of Web pages in terms of font size, language, screen layout, volume control, brightness and contrast. These settings are stored on the user's smart card for retrieval and use during a future access session.

Furthermore, since the user can be positively identified by means of the user's smart card and PIN

verification procedure, personal data, such as E-mails, voice messages and fax messages can be securely forwarded to the user at the user terminal within an hotel room, or to a different selected service provider.

5 Finally, the user's areas of interest can be stored on the smart card. When logging in, one or more Web searches would be initiated to provide details of new information available since last use of
10 the profile. This would be implemented by saving the results of the previous searches and excluding from the current search those entries which have previously been viewed by the user.

Other Embodiments

15 It will be appreciated from the above description that the present invention in its various embodiments provides for an enhanced speed of response for improved convenience and acceptability of a Web access service provided within a commercial environment, such
20 as hotel rooms. The present invention may also be applied in other commercial environments, such as within hospitals and in cruise liners. In the case of cruise liners, the low-level cache stores 102 may be linked to the remainder of the system via satellite
25 links.

It should be noted that, although the invention

thus far has been explained in relation to embodiments having a proxy cache server 6 which is separate from a client terminal 2, the caching procedure implemented in the cache server 6 could also be implemented in a client terminal 2, to realise a client terminal having efficient automatic pre-fetching capabilities.

In addition, although not yet mentioned, the client 2 may itself cache, for the duration of one user session, Web pages access during that user session. Such functionality is already provided when using a Netscape browser, and reduces the need for constant refreshing of page data within the client 2 during a user session.

It should furthermore be appreciated that various equivalents, modifications and variations can be employed in relation to the features described in each of the preferred embodiments, without departing from the spirit or scope of the present invention.

CLAIMS:

1. A caching system for requesting data files from a server network, in which said data files are identified by file addresses, and for storing same prior to retrieval via said system by a client, said
5 system being configured to:

- (i) access a dynamic set of file addresses;
- (ii) request and store data files having file addresses appearing in said set;
- 10 (iii) assess the likelihood of retrieval of a data file via said system in accordance with predefined criteria; and
- (iv) modify said set in accordance with said assessed likelihood.

15 2. A caching system according to claim 1, and configured to remove a file address from said set when the retrieval of a corresponding data file via said system is relatively infrequent.

20 3. A caching system according to claim 2, and configured to remove said file address after said corresponding data file is not retrieved via said system for a predetermined period of time.

4. A caching system according to any of claims

1 to 3, and configured to add a file address to said set when a corresponding data file is retrieved via said system.

5 5. A caching system according to any of claims 1 to 4, and configured to automatically request and store an updated data file having a file address appearing in said set, for which address a data file was already stored, without reference to said stored data file.

10 6. A caching system according to claim 5, and configured to store a timing parameter for a file address appearing in said set, said automatic request being made in accordance with said timing parameter.

15 7. A caching system according to claim 6, and configured to verify via said server network that a stored data file is current when being retrieved by a client, and to modify said timing parameter if said stored data file is not current.

20 8. A caching system according to claim 6 or 7, wherein an individual such timing parameter is stored for each file address appearing in said set.

9. A caching system for requesting data files from a server network, in which said files are identified by file addresses, and for storing same prior to retrieval via said system by a client, said system being configured to:

- (i) automatically request a current version of a data file for a file address appearing in a predefined set of file addresses, for which address a data file was already stored, without reference to said stored data file;
- (ii) request a current version of a data file for said file address when being retrieved by a client; and
- (iii) store said current version when received from said server network, for subsequent retrieval by a client.

10. A caching system according to claim 9, and configured to store a timing parameter for a file address appearing in said set, said automatic request being made in accordance with said timing parameter.

11. A caching system according to claim 10, and configured to verify via said network that a stored data file is current when being retrieved by a client,

and to modify said timing parameter if said stored data file is not current.

12. A caching system according to any of claims 9 to 11, wherein said set comprises members selected
5 in response to the retrieval of corresponding data files via said system.

13. A caching system according to any of claims 9 to 12, wherein said set comprises members selected independent of retrieval of a corresponding data file
10 via said system.

14. A caching system for requesting data files from a server network, in which said files are identified by file addresses, and for storing same prior to retrieval via said system by one or more
15 clients, said system being configured to automatically request a current version of a data file for a file address appearing in a predefined set of file addresses, said set comprising members selected in response to the retrieval of corresponding data files
20 via said system, and members selected independent of retrieval of a corresponding data file via said interface.

15. A caching system for requesting data files from an external server network, in which said data files are identified by file addresses, and for storing same prior to retrieval via said system by a client, said system comprising a plurality of low-level cache servers, each said low-level cache server respectively having an associated plurality of clients, and a higher-level cache server serving said plurality of low-level cache servers, said system being configured such that each of said low-level cache servers stores an individual set of data files, and said higher-level cache server stores data files appearing in each of said individual sets.

16. A caching system according to claim 15, wherein each of said low-level servers is located at a respective local access facility, such as an hotel, a cruise ship or a hospital.

17. A caching system according to claim 15 or 16, comprising a plurality of said higher-level cache servers, serving pluralities of said low-level cache servers in different geographical regions, said higher-level cache servers storing different sets of data files in accordance with the data files stored in the respective low-level cache servers.

18. A caching system according to claim 18, further comprising an even-higher-level server storing data files appearing in each of said different sets.

5 19. A caching system according to any of claims 15 to 18, wherein a client is able to retrieve data files from one of each of said types of server, or from the external server network.

10 20. A caching system according to any of claims 15 to 20, wherein each said server sends to said external server network a conditional request for a current version of a data file when receiving a request for said data file of which a version is stored, and provides said stored version when receiving confirmation that the stored version is
15 current.

20 21. A caching system for requesting data files from a server network, in which said data files are identified by file addresses, and for fetching and storing same prior to retrieval by a user, said system comprising user profile storage means, accessible by the system, said user profile comprising a set of data file addresses to be fetched and stored by said system for subsequent retrieval.

22. A caching system according to claim 21, wherein said user profile further comprises screen presentation settings specified by said user for use during presentation of said data files.

5 23. A caching system according to any preceding claim, wherein said data files are Word-Wide Web pages.

10 24. A caching substantially as hereinbefore described, in particular with reference to any of the accompanying drawings.



Patent
Office

36

Application No: GB 9620665.1
Claims searched: 1-8, 23, 24

Examiner: B G Western
Date of search: 13 November 1996

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.O): G4A AMC AMU

Int Cl (Ed.6): G06F 12/08 17/30

Other: On-line databases: COMPUTER, INSPEC, WPI

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
A	EP-0604013-A2 SUN MICROSYSTEMS See whole document	-
A	US-5305389-A PALMER See whole document	-
A	US-4897781-A CHANG et al See whole document	-
X	WO-95/21415-A1 UNIVERSITY OF CALIFORNIA N.b. pages 15-21	1-6,8

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.

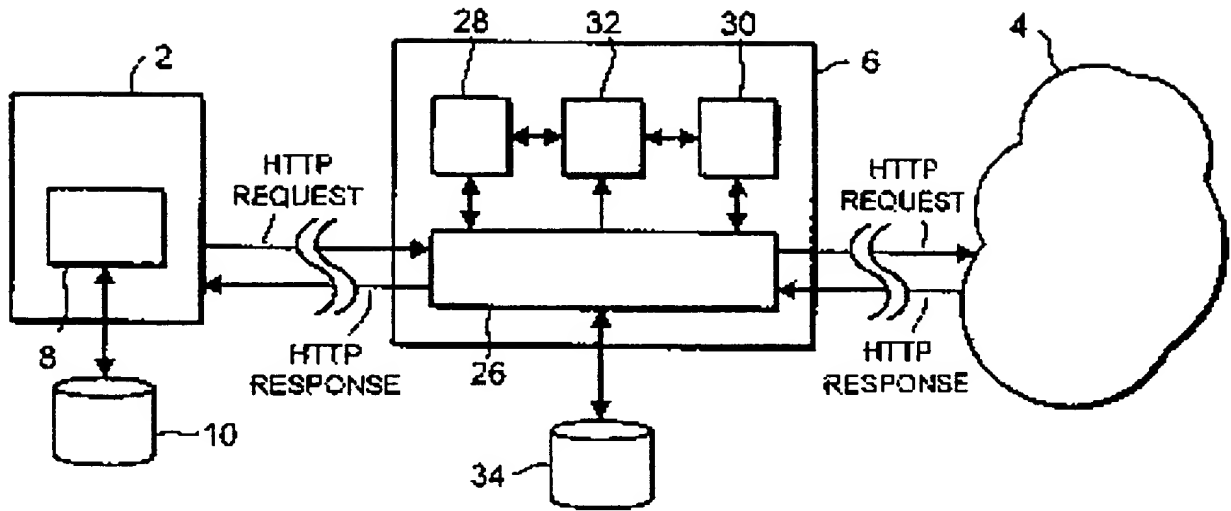


FIG. 1

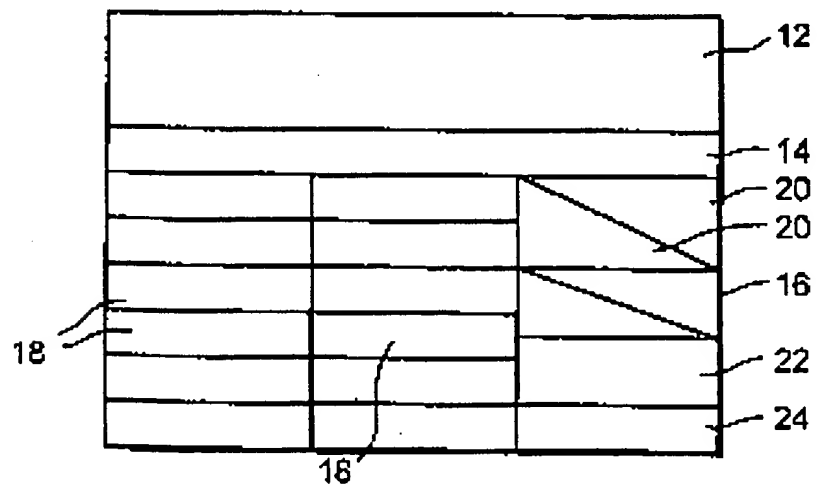


FIG. 2

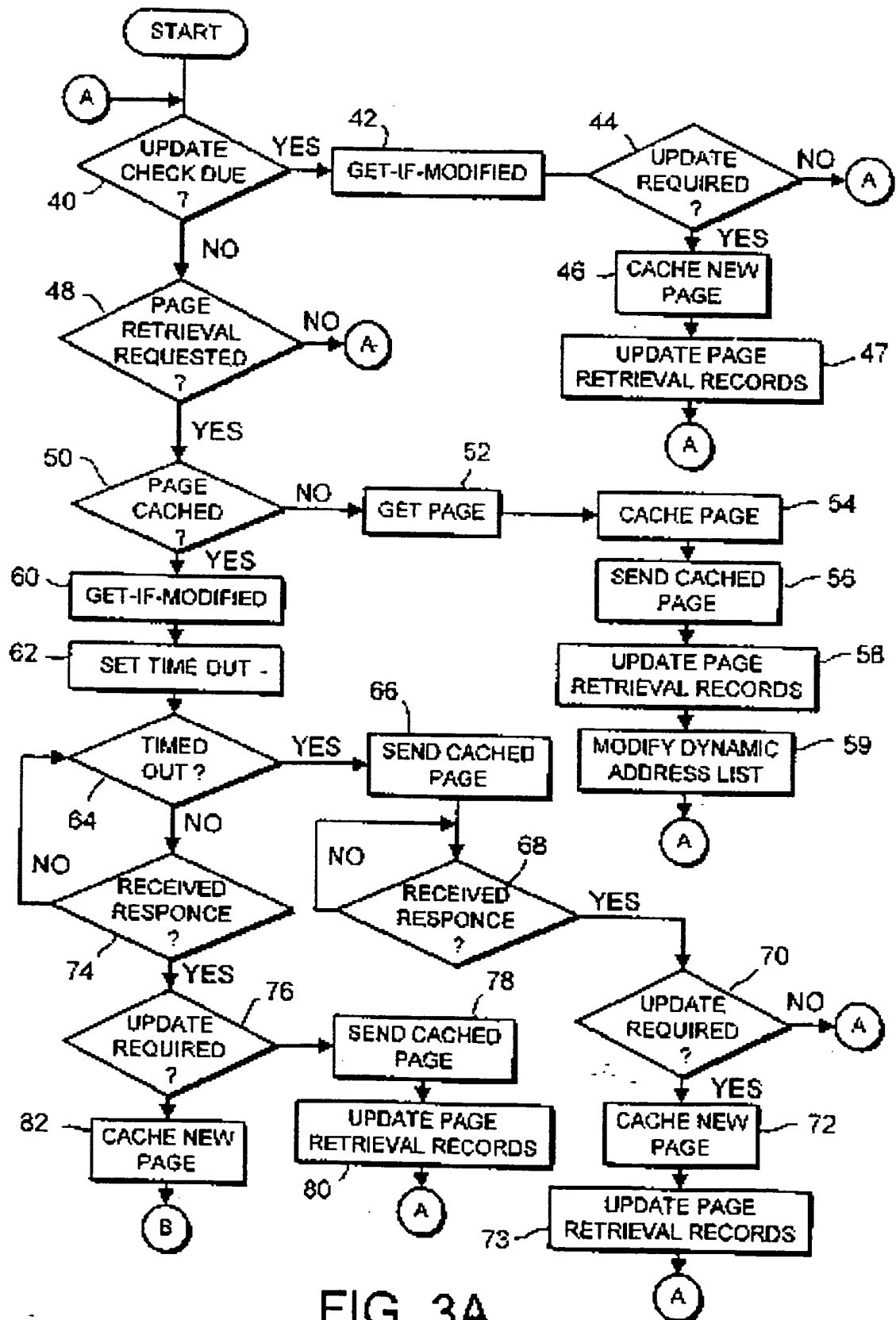


FIG. 3A

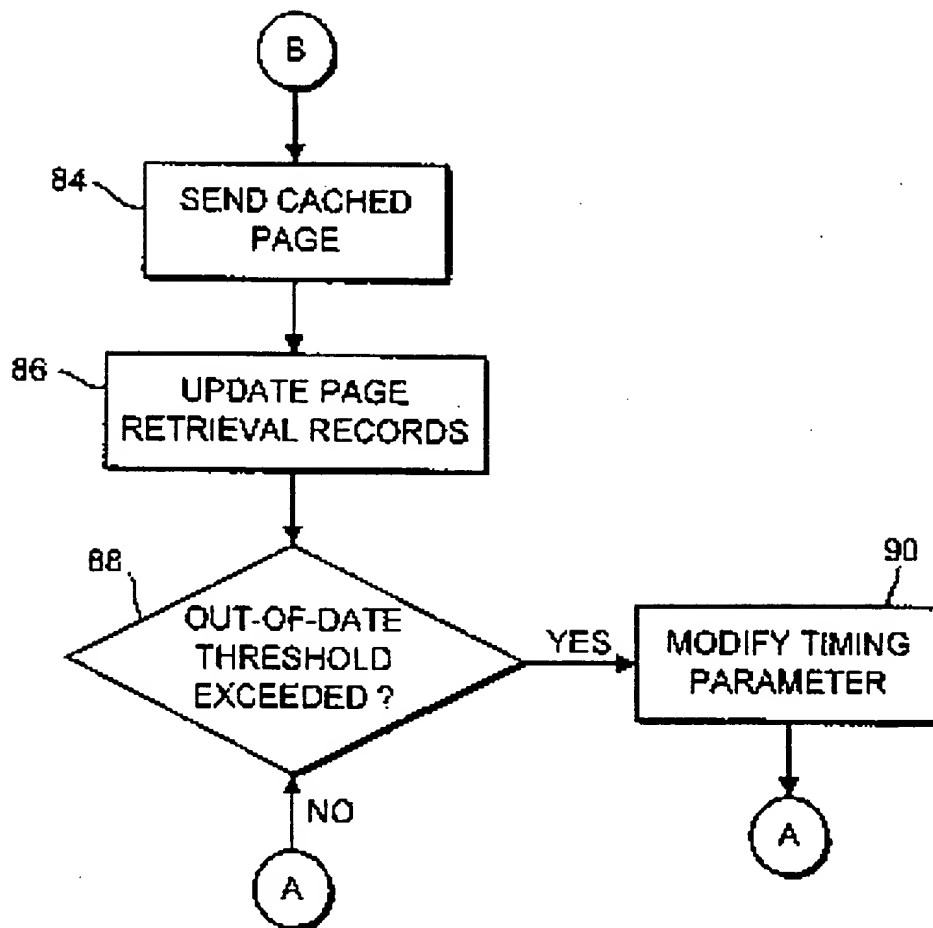


FIG. 3B

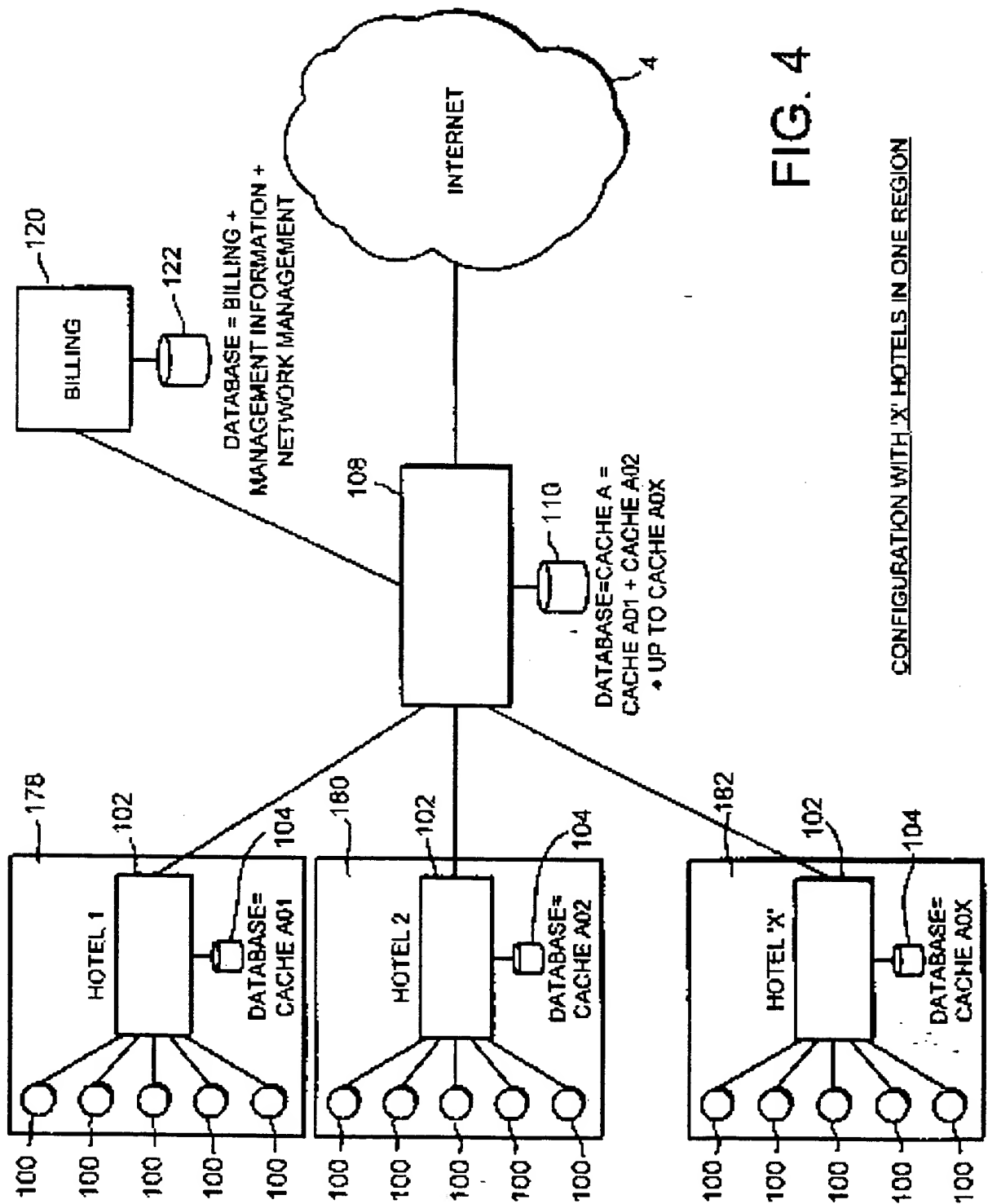


FIG. 4

CONFIGURATION WITH 'X' HOTELS IN ONE REGION

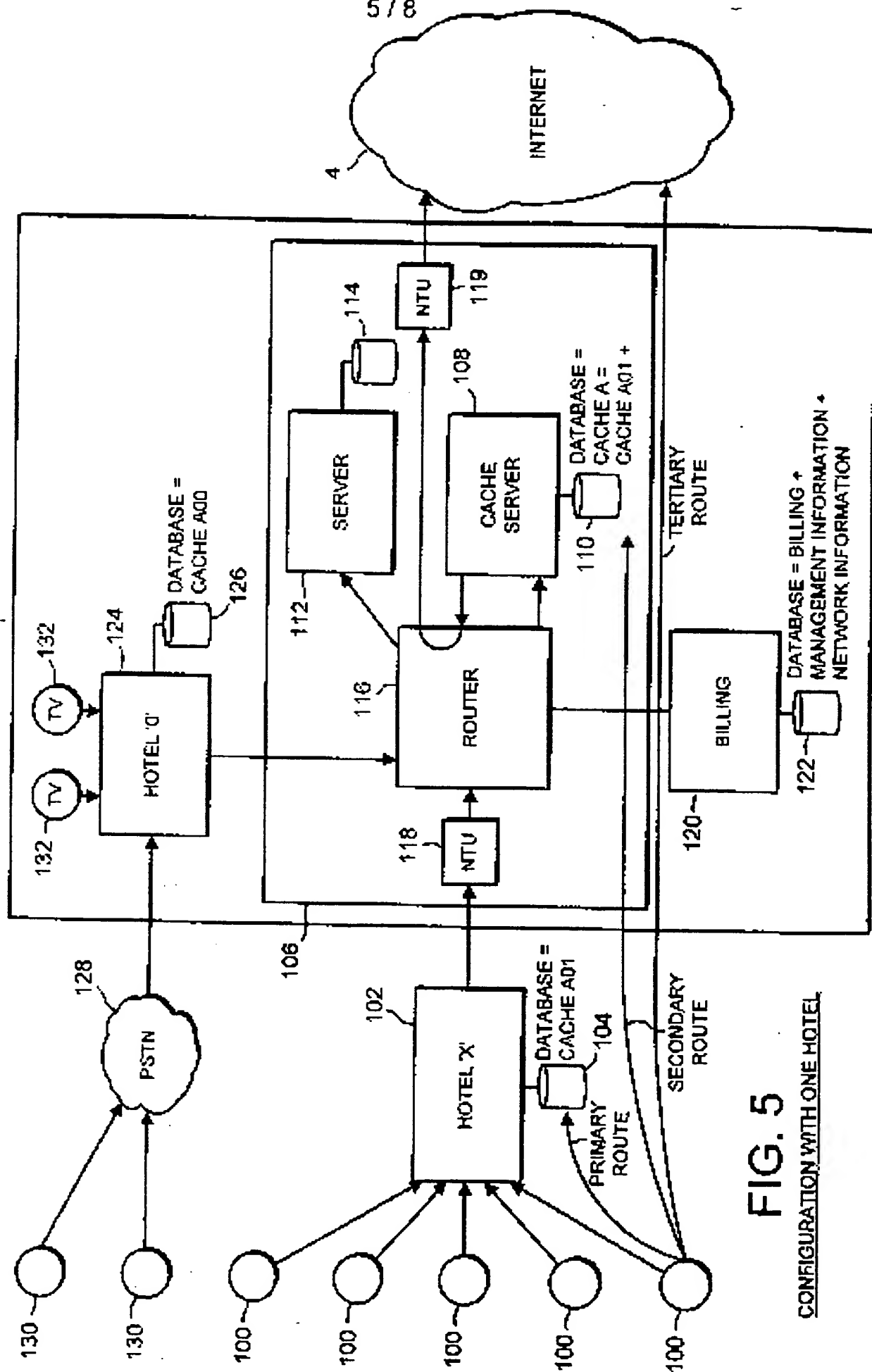


FIG. 5

CONFIGURATION WITH ONE HOTEL

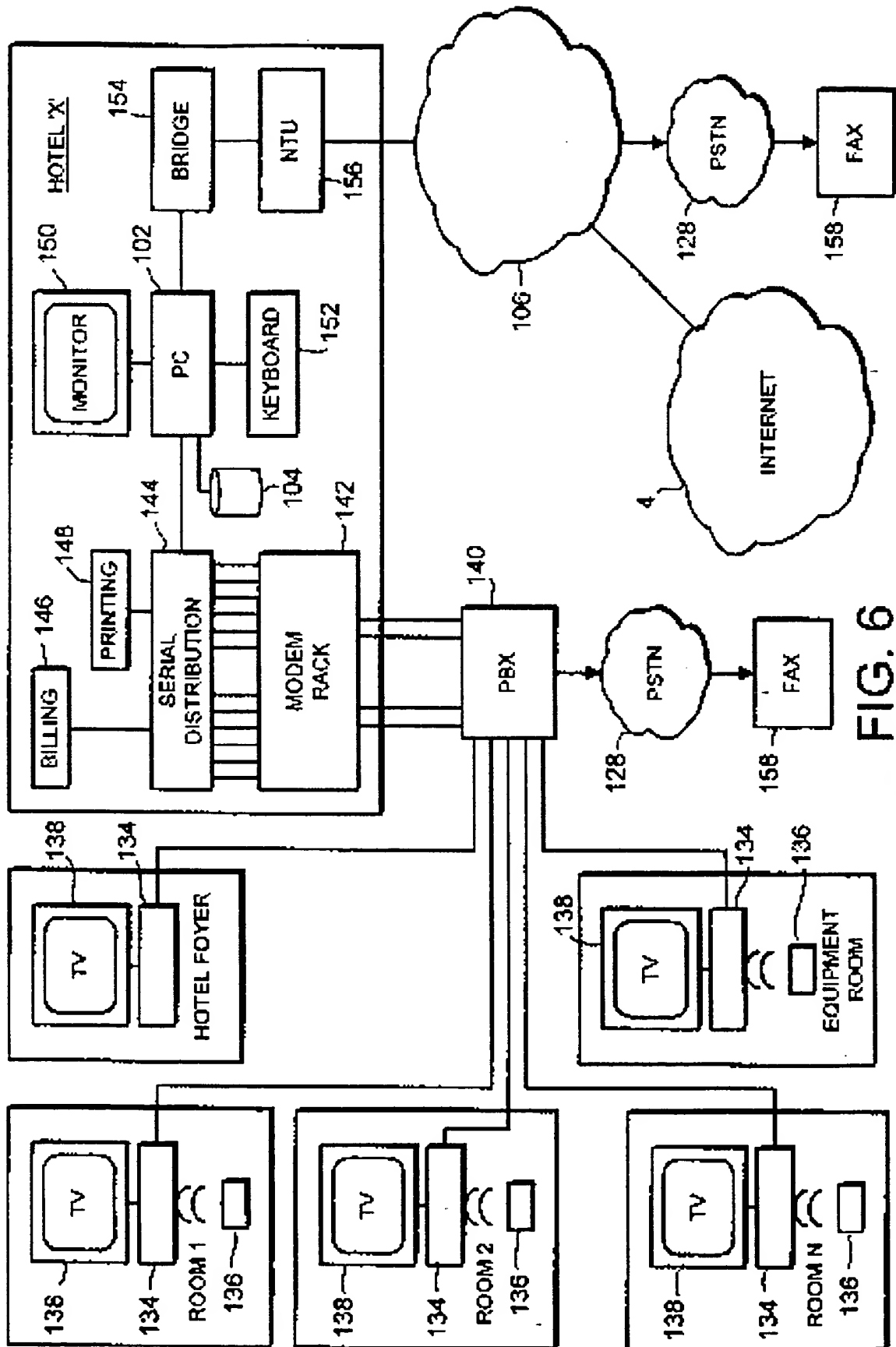


FIG. 6
HOTEL CONFIGURATION

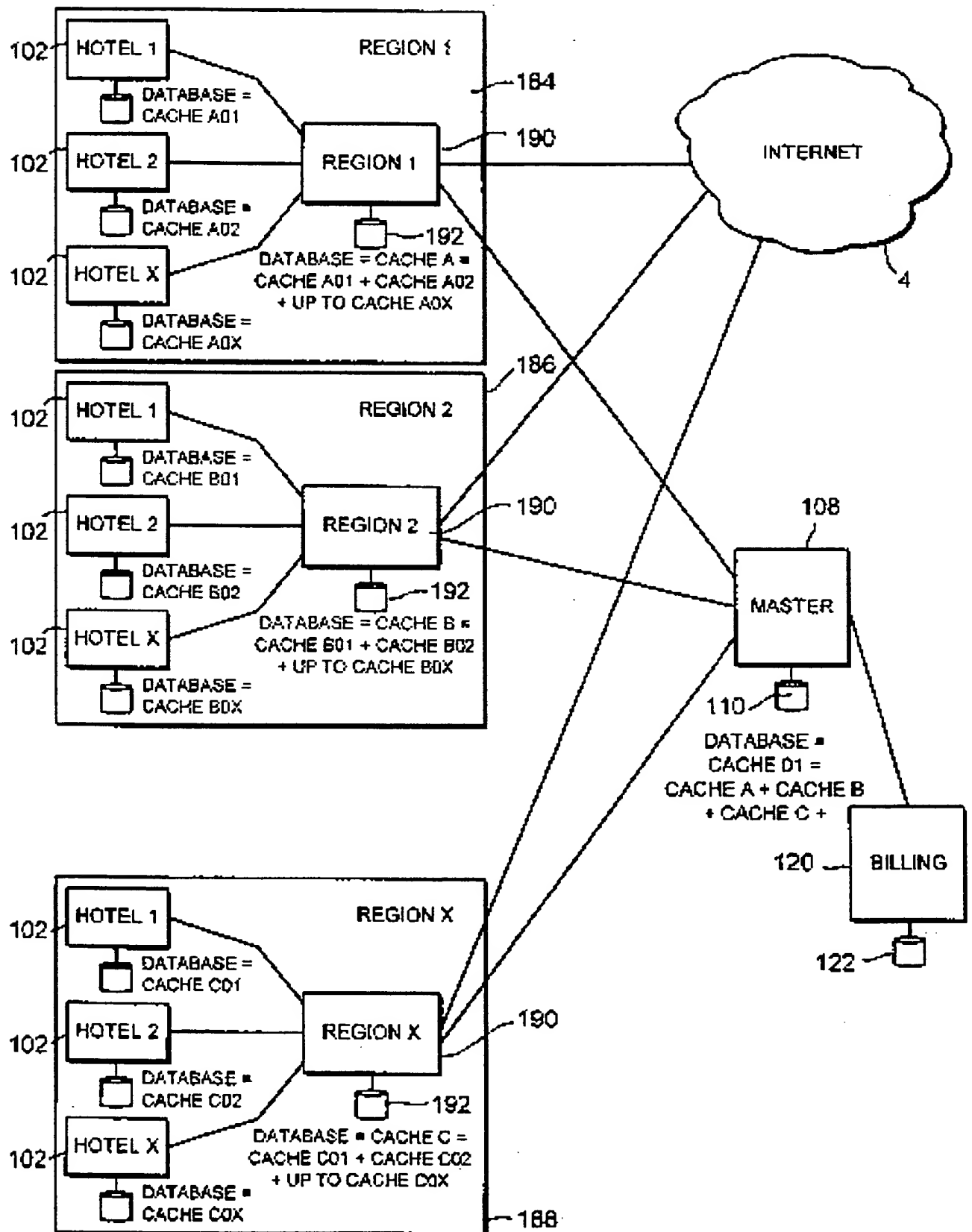


FIG. 8

CONFIGURATION WITH 'X' HOTELS IN 'X' REGIONS